

## RSM-1: A MACHINE LANGUAGE MONITOR FOR THE RADIO SHACK COMPUTER

RSM-1 is a powerful program that allows you to interact directly with your Radio Shack computer at the machine language level. It will require a certain amount of time to become familiar with RSM-1, but don't be afraid to experiment. Load the program using the RADIO SHACK CLOAD command. If a satisfactory load is achieved, the screen will be cleared, the copyright notice will be displayed, and the screen will respond with:

### COMMAND?

This means the monitor is awaiting input from the keyboard. Entries will be written from top to bottom on the screen, and will scroll off the top when the screen is full. You may clear the screen with the CLEAR key. Now type D followed by ENTER. The screen will fill with lines of hex characters, representing memory starting at location 0. Hit the space bar to stop the scroll. Step through several lines with the space bar. Type any other key and the scrolling will continue. Type BREAK (or shift C) and COMMAND? will reappear. Try the same thing but use A instead of D. Allow the dump to continue until you see addresses in the 01B0 range. Stop the scroll and step line by line with the space bar. You are now looking at the error messages in your BASIC ROM's. You will find the BASIC command table up around 0250. Now read the following detailed description to find out what else RSM-1 can do. Experiment as you go, you can't hurt your computer from the keyboard. At worst, you may have to reload your RSM-1 cassette.

### COMMAND FORMAT

All commands are single letter commands, and may have up to three 16-bit hex addresses following the command. Addresses are separated by spaces. The first space is optional. The following commands are all valid commands:

A	No addresses are required.
A 31A0	31A0 is starting address.
D243F 4024	First space is optional.
M 1234 4365 E080	Three fields are maximum.
D 1 10	Leading zeros are optional.
Z 433621543A 8876680	Equivalent to Z 543A 6680.

Several points are illustrated in the above sequence:

- 1) Commands do not require addresses.
- 2) Leading spaces are optional.
- 3) Leading zeros are optional.
- 4) If more than 4 hex address characters are entered, only the last four are valid.

The following command features are not readily illustrated:

- 5) For most commands, if addresses are not entered, default values will be assumed.
- 6) All commands must be started with a valid letter. If an invalid letter is entered, it will be ignored.
- 7) All addresses are in hex. If a non-hex character is entered in an address field, it will be ignored.
- 8) Spaces are used to separate the address fields. Only one space is allowed; additional spaces are ignored.
- 9) Any command may be aborted by the BREAK key.
- 10) Any character may be erased by a LEFT-ARROW.

PAUSE - Any command that might use a significant amount of time may be interrupted by entering a space. This is usually used in conjunction with commands that dump memory onto the screen. A second space will cause one line of text to be dumped, with a pause again at the end of the line. Typing any character except a space will cause the dump to continue without interruption. The BREAK key will abort any routine and return control to the monitor.

#### COMMAND EXECUTION

All commands are executed by the ENTER key. Commands may be edited by backspacing (LEFT-ARROW) or may be aborted with the BREAK key up until the time ENTER is typed.

#### DETAILED COMMAND DESCRIPTION

Commands may be separated into three groups: Control, Tape, and Memory. They may also be assigned a number corresponding to the number of address fields that are effective. If a command needs more addresses than are input, default values will be assigned. If more addresses are input than required, the first addresses will be used; extras will be ignored.

#### CONTROL COMMANDS

G - GOTO: 1 address. Transfers control to specified address. If no addresses are entered, control returns to BASIC.

I - INITIALIZE: 0 addresses (See INPUT command below). Clears screen and initializes all scratchpad locations. This command is accessed automatically after loading RSM-1, and thereafter when I is typed, or if RSM-1 is re-entered at address 4200 hex.

I - INPUT FROM PORT: 1 address. If I is typed followed by a hex address, the input port at the specified address will be read, and the hex value displayed on the screen. The stock Radio Shack computer has one input port (cassette) at address FF.

O - OUTPUT TO PORT: 2 addresses. The O command will output a specified hex byte to a specified output port. The first address is the output port, and the second is the byte that will be output. If no addresses are typed, RSM-1 will output a 0 to port 0. If only one address is typed, RSM-1 will output a 0 to the specified port.

U - USER COMMAND: 0-3 addresses. The U command allows the executive routines of RSM-1 to be used in conjunction with custom commands or programs. When U is executed, program control is transferred to address 4880 hex, where the user's program should be located. If a normal return is encountered at the end of the user routine, control will revert to RSM-1. If addresses are entered with the U command, the first address will be found in the D,E registers, the second in H,L and the third in B,C. During RSM-1 initialization, a return instruction (C9 hex) is placed at 4880. Thus if U is typed accidentally, no harm is done as control returns to RSM-1.

K - KEYBOARD ECHO: 0 addresses. Allows characters to be typed directly from the keyboard to the screen.

B - BINARY ARITHMETIC: 1-2 addresses. This utility routine is useful for converting hex addresses to decimal. If one address is input, the decimal equivalent is output. If two addresses are input, a title line, both addresses, A+B, A-B, and B-A are printed both in hex and in decimal.

#### TAPE COMMANDS

Cassette tapes read in two modes, one allowing entry of tape length and starting address, and a second mode with self-locating capabilities. The 'standard' format is used to read in tapes written by programs other than RSM-1, such as BASIC tapes. Header tapes, which are written by RSM-1, contain the tape length and starting location as the first four bytes on the tape, thus it is not necessary to keep track of this information after tapes are written. Header tapes may be loaded into their original locations by a single letter command, or may be located anywhere in memory since a single input address will override the header starting location. Header tapes use checksum error detection and may be checked without disturbing memory.

Both read and write operations commence with the ENTER key. After reading and writing operations are complete, the tape checksum is output onto the screen and the COMMAND? string is output again. Either operation may be aborted with the BREAK key.

#### TAPE WRITE COMMAND

W - WRITE TAPE: 2 addresses. Writes cassette tape with header format. Writes checksum at end of tape. Addresses are start and end memory locations. Tape writing may be aborted by use of the BREAK key. After tape writing is complete, the checksum will be written onto the screen. Note: Tapes written by RSM-1 will not load using CLOAD, but must be read using the RSM-1 R or L commands.

## TAPE INPUT COMMANDS

R - READ TAPE: 0-2 addresses. Loads either of two tape formats into memory, controlled by input addresses. If no addresses are input, the monitor assumes a header tape and reads the first four bytes from the tape as the length and start locations. If one address is input, the header format is assumed, but only the length is used; the typed address is used for the start location. If two addresses are input, the monitor will read the tape in using the first address as the tape length and the second address as the starting location in memory. If a tape error is detected, the monitor outputs CHECKSUM ERROR followed by two hex bytes, the checksum from the tape and the computed checksum.

L - LOAD AND GO: 0-2 addresses. Operates the same as R, but program control is transferred to the start address of the loaded program at end of load. If an error is detected, the CHECKSUM ERROR message is output and control is not transferred.

C - CHECK TAPE: 0-2 addresses. Operates the same as R, but the program is not loaded into memory. Used primarily after writing a tape to verify accuracy. Computes checksum from information read from tape and compares it with checksum recorded on tape.

## MEMORY COMMANDS

Memory may be displayed, modified, moved, exchanged, or tested. All memory display commands may be stepped through one line at a time by using the space bar, or terminated by use of BREAK.

A - ASCII DUMP: 0-2 addresses. Interprets and displays memory as ASCII characters. Non-ASCII characters and control characters are displayed as periods. Each line starts with the address of the first location displayed, followed by 48 characters, unless a carriage return is encountered, which will cause a new line to be started. This feature formats ASCII files for easy reading.

Addressing: The first and last addresses are the starting and ending locations respectively. If no addresses are input, dump will start at 0 and continue until aborted. One address will cause a continuous dump starting at the typed address. Two addresses will dump memory from the first address through the second address. Space bar will stop dump at the end of the current line; additional spaces will output single lines. Any other character will cause the continuous dump to resume.

A SSSS 0 - FORMATTED ASCII DUMP. If two addresses are input and the second is 0, a formatted ASCII dump is output. This command lists memory with 16 characters per line, starting at address SSSS. Printable ASCII characters are listed normally. If the 8th bit of a printable character is set, the character will be followed by a period. If the character is non-printing, the hex value of the character will be printed.

D - HEX DUMP: 0-2 addresses. Interprets and displays memory as two hex characters per byte. Start address is displayed at the beginning of each line; 16 bytes are displayed per line. Addressing and pause routines as in "A" command.

**E - EDIT MEMORY:** 1 address. Displays address and hex value of input address followed by a dash (-). This location may be modified by entering two hex characters. If more than two characters are entered, the last two will be used (the back-arrow is not active for this command). Thus errors may be corrected by retyping the correct values. Entering a space will transfer the screen entry to memory and display the following memory location. If nothing is typed after the dash, the spacebar will step to the next location without changing the memory location. The ENTER key will have the same effect, but a new line and new address will be displayed on the screen. Typing an up-arrow will back up one byte, and a new line with the new address will be output. Exit the routine with BREAK, which will also transfer the last displayed byte into memory.

**F - FIND 1 BYTE:** 3 addresses. Searches through memory block defined by the first two addresses and finds all locations with the byte specified in the third address field. Writes address, the located byte, and the following byte (in hex) on the screen. This command is designed to locate input/output commands with their arguments. It can also be used for other searches, of course, with the proper interpretation of the output arguments.

**H - HUNT 2 BYTES:** 3 addresses. Searches memory from first to second address for two-byte combinations specified by third address. Address is interpreted in 'reverse' notation (low byte first). Address of location preceding the found address is output, followed by the code and finally followed by the address. This command is designed to locate references to memory locations and to identify the referencing command, but may be used for other searches, provided 'reverse' entry is used for the desired two-byte combination.

**M - MOVE MEMORY:** 3 addresses. Moves a block of memory between the first and second addresses to a similar block starting at the third address. There are no restrictions on addressing - overlap from either end is allowed.

**Q - CHECKSUM:** 2 addresses. Computes and displays checksum for memory block between two addresses. Used to determine if program has been 'bombed' during operation.

**T - TEST MEMORY:** 2 addresses. Uses random numbers to test memory between two specified addresses. Writes address, expected value and actual value for locations where errors occur. Writes number of tests completed as they are finished; 256 checks of each location is defined as 1 test.

**V - VERIFY MEMORY:** 3 addresses. Compares block of memory from first address to second address with block starting at third address. Writes address and both values where differences occur.

**X - EXCHANGE MEMORY:** 3 addresses. Exchanges block of memory between first and second addresses with block starting at third address.

**Z - ZERO MEMORY:** 2-3 addresses. Writes code in third address into block of memory between first two addresses. For two addresses, 00 is written into the block between the addresses. As with all commands, addresses are inclusive.

## ADDITIONAL FEATURES

## JUMP TABLE

RSM-1 has three jumps located at the beginning of the program at addresses 4203, 4206, and 4209 (the first three bytes at 4200 are set to 00 during initialization). The first jump is to the initialization routine and is used for re-entering RSM-1. The second jump, at 4206, is to the CRT output routine. You may use this location to write directly to the screen, with normal scrolling, etc. The character to be output should be in the A register. If you use a form feed (hex 0C) character, the screen will be cleared. The third jump at 4209 is to the keyboard routine. If you call this location, the keyboard routine will not return until a key has been pressed. The typed character will be returned in the A register.

## SOURCE LISTING

There are many other routines in RSM-1 that might be useful for machine language programs. A source listing (using INTEL 8080 mnemonics) is available for \$5.00. You may order the source when you send in the registration form, which will also place you on our mailing list for further announcements of Radio Shack compatible products, or you may order the source at a later date. However, only registered RSM-1 owners may purchase the source listing.

---

SMALL SYSTEM SOFTWARE  
BOX 483 Newbury Park  
California 91320

RADIO SHACK MONITOR "RSM-1" COMMAND SUMMARY

The following list of commands is included in the monitor. Many commands have several variations, depending upon the number of address fields following the command letter.

- A - ASCII DUMP: Display ASCII equivalent of memory block.
- B - BINARY ARITHMETIC: Add/subtract in hex and decimal.
- C - CHECK TAPE: Check cassette tapes for proper checksum.
- D - HEX DUMP: Display Hex equivalent of memory block.
- E - EDIT: Enter, examine, or modify memory in hex code.
- F - FIND 1 BYTE: Find and display single-byte hex codes.
- G - GOTO: Go to and execute program at specified address.
- H - HUNT 2 BYTES: Find and display two-byte addresses.
- I - INITIALIZE/INPUT: Initialize or input data from port.
- K - KEYBOARD ECHO: Type directly to screen or terminal.
- L - LOAD AND GO: Load cassette tape and execute program.
- M - MOVE: Move any block of memory to specified location.
- O - OUTPUT: Output hex value to specified output port.
- Q - CHECKSUM: Compute checksum of specified memory block.
- R - READ TAPE: Read cassette tape (header or standard).
- T - TEST MEMORY: Test memory block and display errors.
- U - USER: Allows user to write and execute new commands.
- V - VERIFY MEMORY: Compare any two blocks of memory.
- W - WRITE TAPE: Write any memory block to cassette tape.
- X - EXCHANGE: Interchange any two blocks of memory.
- Z - ZERO MEMORY: Write zero or any hex code into memory.

```

4200      0010 *          * * * * *
4200      0020 *          * RSM-1      VERSION 1.0 *
200      0030 *          * SMALL SYSTEM SOFTWARE *
4200      0040 *          *      TOM NUSSMEIER      *
4200      0050 *          *      GARY THURMOND      *
4200      0060 *          * COPYRIGHT (C) FEB. 1978 *
4200      0070 *          * ALL RIGHTS RESERVED *
4200      0080 *          * * * * *
4200      0090 *
4200      0100 *
4200      0110 *
4200      0120 * BASIC DEFINITIONS FOR ASSEMBLY
4200      0130 *
4200      0140 MONI EQU $+3 MONITOR LOC.
4200      0150 SCPAD EQU $-100H SCRATCHPAD
4200      0160 RAM EQU 4880H USER LOCATION
4200      0170 *
4200      0180 *
4200      0190 *
4200      0200 * DEFINITIONS AND CONSTANTS
4200      0210 *
4200      0220 STAT EQU 387FH KEYBOARD PORT
4200      0230 SCAN EQU 0B55H
4200      0240 RLESE EQU 0B4CH
4200      0250 CIN1 EQU 0F81H TAPE LINKS
4200      0260 CIN8 EQU 0FA1H
4200      0270 COUT8 EQU 0FA9H
4200      0280 BUFL EQU 62 BUFFER LENGTH
4200      0290 *
4200      0300 *
4200      0310 *
4200      0320 ORG SCPAD SCRATCHPAD
4100      0330 *
4100      0340 STAK EQU $+100H
4100      0350 IBUF DS 48 INPUT BUFFER
4130      0360 IFLAG DS 1 INITIALIZATION FLAG
4131      0370 FLGP DS 1 PAUSE FLAG
4132      0380 TCNTR DS 1 MEMORY TEST COUNTER
4133      0390 DECNB DS 1 MEMORY TEST NUMBER
4134      0400 YTEMP DS 2 HEX MATH STORAGE
4136      0410 RDPTR DS 2 MEMORY READ POINTER
4138      0420 WRPTR DS 2 MEMORY WRITE POINTER
413A      0430 GOLOC DS 2 GO COMMAND STORAGE
413C      0440 *
413C      0450 *
413C      0460 *
413C      0470 ORG MONI PROGRAM START POINT
4203      0480 *
4203      0490 *
4203      0500 * JUMP TABLE FOR I/O LINKAGE
4203      0510 *
4203 C3 16 43      0520 JMP INIT MONITOR ENTRY
206 C3 7C 44      0530 JMP CRT DISPLAY CHARACTER
209 C3 83 42      0540 JMP KEYB READ KEYBOARD
420C      0550 DS 9 JUMP RESERVE
4215      0560 *

```



4215	2C	0570	CMSG	DB	44	SIGN-ON MESSAGE
4216	52 53 4D 2D 31	0580		ASC	'RSM-1 V-1.0	(C) 1978'
	20 56 2D 31 2E					
	30 20 20 28 43					
	29 20 31 39 37					
	38					
422B	0D	0590		DB	13	
422C	53 4D 41 4C 4C	0600		ASC	'SMALL SYSTEM SOFTWARE'	
	20 53 59 53 54					
	45 4D 20 53 4F					
	46 54 57 41 52					
	45					
4241	0D	0610		DB	13	
4242		0620	*			
4242	3A 7F 38	0630	KEYBG	LDA	STAT	KEYBOARD ROUTINE,
4245	B7	0640		ORA	A	NO STATUS WAIT
4246	C8	0650		RZ		
4247	D9	0660		DB	0D9H	
4248	CD 55 0B	0670		CALL	SCAN	
424B	D9	0680		DB	0D9H	
424C	CD 4C 0B	0690		CALL	RLESE	
424F	E6 7F	0700		ANI	7FH	
4251	C9	0710		RET		
4252		0720	*			
4252	CD 42 42	0730	KEYBC	CALL	KEYBG	KEYBOARD WITH
4255	C8	0740		RZ		BREAK CHECK
4256	FE 03	0750		CPI	3	
4258	C0	0760		RNZ		
4259	C3 38 43	0770		JMP	PRMPT	
425C		0780	*			
425C	CD 74 42	0790	CIN	CALL	PAUSE	TAPE SUBROUTINES
425F	C5	0800		PUSH	B	
4260	CD A1 0F	0810		CALL	CIN8	
4263	C1	0820		POP	B	
4264	C9	0830		RET		
4265		0840	*			
4265	06 80	0850	PREP	MVI	B,80H	
4267	AF	0860		XRA	A	
4268	CD 71 42	0870		CALL	COUT	
426B	05	0880		DCR	B	
426C	C2 67 42	0890		JNZ	PREP+2	
426F	3E A5	0900		MVI	A,0A5H	
4271	CD A9 0F	0910	COUT	CALL	COUT8	
4274	3A 31 41	0920	PAUSE	LDA	FLGP	MONITOR PAUSE
4277	FE 20	0930		CPI	' '	
4279	CA 83 42	0940		JZ	KEYB	
427C	CD 52 42	0950		CALL	KEYBC	
427F	C8	0960		RZ		
4280	FE 20	0970		CPI	' '	
4282	C0	0980		RNZ		
4283	CD 52 42	0990	KEYB	CALL	KEYBC	KEYBOARD WITH
4286	CA 83 42	1000		JZ	KEYB	STATUS WAIT
4289	32 31 41	1010		STA	FLGP	
428C	C9	1020		RET		
428D		1030	*			
428D	FE 30	1040	HEXC	CPI	'0'	HEX CHECK
428F	D8	1050		RC		
4290	FE 3A	1060		CPI	'9'+1	

4292	3F			1070	CMC			
4293	D0			1080	RNC			
4294	FE	41		1090	CPI	'A'		
4296	D8			1100	RC			
4297	FE	47		1110	CPI	'F'+1		
4299	3F			1120	CMC			
429A	C9			1130	RET			
429B				1140	*			
429B	29			1150	AHEX	DAD	H	ASCII TO BINARY
429C	29			1160		DAD	H	
429D	29			1170		DAD	H	
429E	29			1180		DAD	H	
429F	D6	30		1190		SUI	48	
42A1	FE	0A		1200		CPI	10	
42A3	DA	A8	42	1210		JC	\$+2	
42A6	D6	07		1220		SUI	7	
42A8	85			1230		ADD	L	
42A9	6F			1240		MOV	L,A	
42AA	C9			1250		RET		
42AB				1260	*			
42AB	46			1270	STRNG	MOV	B,M	PRINT STRING
42AC	23			1280		INX	H	
42AD	7E			1290		MOV	A,M	
42AE	CD	7C	44	1300		CALL	CRT	
42B1	05			1310		DCR	B	
42B2	C2	AC	42	1320		JNZ	STRNG+1	
42B5	C9			1330		RET		
42B6				1340	*			
42B6	3E	0D		1350	CRLF	MVI	A,13	
42B8	CD	7C	44	1360		CALL	CRT	
42BB	3E	0A		1370		MVI	A,10	
42BD	C3	7C	44	1380		JMP	CRT	
42C0				1390	*			
42C0	FE	03		1400	DLMT	CPI	3	DUMP LIMIT
42C2	DA	C8	42	1410		JC	\$+3	
42C5	21	FF	FF	1420		LXI	H,-1	
42C8	EB			1430	LENG	XCHG		LENGTH COMPUTATION
42C9	7B			1440		MOV	A,E	
42CA	95			1450		SUB	L	
42CB	5F			1460		MOV	E,A	
42CC	7A			1470		MOV	A,D	
42CD	9C			1480		SBB	H	
42CE	57			1490		MOV	D,A	
42CF	C9			1500		RET		
42D0				1510	*			
42D0	06	05		1520	ERR	MVI	B,5	OUTPUT "ERROR"
42D2	21	A5	47	1530		LXI	H,CERR+9	
42D5	C3	AC	42	1540		JMP	STRNG+1	
42D8				1550	*			
42D8	CD	B6	42	1560	ADRO	CALL	CRLF	OUTPUT ADDRESS
42DB	CD	EB	42	1570		CALL	HLOUT	
42DE	3E	3A		1580		MVI	A,':'	
42E0	CD	7C	44	1590		CALL	CRT	
42E3	CD	E6	42	1600	SP2	CALL	SP1	
42E6	3E	20		1610	SP1	MVI	A,' '	
42E8	C3	7C	44	1620		JMP	CRT	
42EB				1630	*			
42EB	7C			1640	HLOUT	MOV	A,H	OUTPUT H,L REGISTERS

```

42EC CD F4 42
42EF 7D
42F0 C3 F4 42
42F3
42F3 7E
42F4 F5
42F5 0F
42F6 0F
42F7 0F
42F8 0F
42F9 CD FD 42
42FC F1
42FD E6 0F
42FF FE 0A
4301 DA 06 43
4304 C6 07
4306 C6 30
4308 C3 7C 44
430B
430B 32 30 41
430E 3A 30 41
4311 FE C3
4313 CA 38 43
4316 21 02 42
4319 36 00
431B 2B
431C 7C
431D FE 40
431F C2 19 43
4322 31 00 42
4325 3E C9
4327 32 30 41
432A 32 80 48
432D 3E 0C
432F CD 7C 44
4332 21 15 42
4335 CD AB 42
4338 31 00 42
433B 21 38 43
433E E5
433F 21 71 44
4342 CD AB 42
4345 CD 83 42
4348 FE 1D
434A CA 45 43
434D FE 0C
434F CA 7C 44
4352 FE 20
4354 DA 45 43
4357 CD 7C 44
435A CD B6 43
435D CA 73 43
4360 21 00 41
4363 01 3E 00
4366 77
4367 23
4368 CD 86 43
436B FE 0D

```

```

1650 CALL OUTH
1660 MOV A,L
1670 JMP OUTH
1680 *
1690 OUTM MOV A,M
1700 OUTH PUSH PSW
1710 RRC
1720 RRC
1730 RRC
1740 RRC
1750 CALL BIASC
1760 POP PSW
1770 BIASC ANI 15
1780 CPI 10
1790 JC NUMB
1800 ADI 7
1810 NUMB ADI '0'
1820 JMP CRT
1830 *
1840 INITC STA IFLAG
1850 START LDA IFLAG
1860 CPI 0C3H
1870 JZ PRMPT
1880 INIT LXI H,STAK+2
1890 INIT2 MVI M,0
1900 DCX H
1910 MOV A,H
1920 CPI <STAK-101H
1930 JNZ INIT2
1940 LXI SP,STAK
1950 MVI A,0C9H
1960 STA IFLAG
1970 STA RAM
1980 MVI A,0CH
1990 CALL CRT
2000 LXI H,CMSG
2010 CALL STRNG
2020 PRMPT LXI SP,STAK
2030 LXI H,PRMPT
2040 PUSH H
2050 LXI H,PR1
2060 CALL STRNG
2070 CMND CALL KEYB
2080 CPI 1DH
2090 JZ CMND
2100 CPI 0CH
2110 JZ CRT
2120 CPI 20H
2130 JC CMND
2140 CALL CRT
2150 CALL COMM
2160 JZ BS1
2170 LXI H,IBUF
2180 LXI B,BUFL
2190 MOV M,A
2200 INX H
2210 CALL READ
2220 CPI 13

```

OUTPUT HEX FORMAT

BINARY TO ASCII

HOT INITIALIZATION  
PROGRAM START

INITIALIZATION

PROGRAM RETURN  
POINT

READ COMMAND

436D CA 17 44  
 4370 C3 45 43  
 4373  
 4373 3E 1D  
 4375 CD 7C 44  
 4378 C3 45 43  
 437B  
 437B 2B  
 437C BE  
 437D 23  
 437E C2 A8 43  
 4381 3E 1D  
 4383 CD 7C 44  
 4386 CD 83 42  
 4389 FE 0C  
 438B CA 86 43  
 438E CD 7C 44  
 4391 77  
 4392 FE 0D  
 4394 C8  
 4395 FE 1D  
 4397 CA B1 43  
 439A FE 20  
 439C CA 7B 43  
 439F DA 86 43  
 43A2 CD 8D 42  
 43A5 DA 81 43  
 43A8 7D  
 43A9 B9  
 43AA CA 81 43  
 43AD 23  
 43AE C3 86 43  
 43B1  
 43B1 2D  
 43B2 C2 86 43  
 43B5 C9  
 43B6  
 43B6 21 C8 43  
 43B9 11 03 00  
 43BC 47  
 43BD 19  
 43BE 7E  
 43BF B7  
 43C0 C8  
 43C1 B8  
 43C2 C2 BD 43  
 43C5 23  
 43C6 5E  
 43C7 23  
 43C8 56  
 43C9 B7  
 43CA C9  
 43CB  
 43CB 41  
 43CC 90 44  
 43CE 42  
 43CF 0A 45  
 43D1 43

2230 JZ DECOD  
 2240 JMP CMND  
 2250 \*  
 2260 BS1 MVI A,1DH  
 2270 CALL CRT  
 2280 JMP CMND  
 2290 \*  
 2300 BLANK DCX H  
 2310 CMP M  
 2320 INX H  
 2330 JNZ FULL  
 2340 BS2 MVI A,1DH  
 2350 CALL CRT  
 2360 READ CALL KEYB  
 2370 CPI 0CH  
 2380 JZ READ  
 2390 CALL CRT  
 2400 MOV M,A  
 2410 CPI 13  
 2420 RZ  
 2430 CPI 1DH  
 2440 JZ ERASE  
 2450 CPI ' '  
 2460 JZ BLANK  
 2470 JC READ  
 2480 CALL HEXC  
 2490 JC BS2  
 2500 FULL MOV A,L  
 2510 CMP C  
 2520 JZ BS2  
 2530 INX H  
 2540 JMP READ  
 2550 \*  
 2560 ERASE DCR L  
 2570 JNZ READ  
 2580 RET  
 2590 \*  
 2600 COMM LXI H,CTBLE-3  
 2610 LXI D,3  
 2620 MOV B,A  
 2630 COM1 DAD D  
 2640 MOV A,M  
 2650 ORA A  
 2660 RZ  
 2670 CMP B  
 2680 JNZ COM1  
 2690 INX H  
 2700 MOV E,M  
 2710 INX H  
 2720 MOV D,M  
 2730 ORA A  
 2740 RET  
 2750 \*  
 2760 CTBLE DB 'A'  
 2770 DW ASCI  
 2780 DB 'B'  
 2790 DW HEXAR  
 2800 DB 'C'

BACKSPACE

LOAD BUFFER

DELETE CHARACTER

COMMAND SEARCH

COMMAND TABLE

43D2 36 47  
 43D4 44  
 43D5 AB 45  
 43D7 45  
 43D8 C9 45  
 43DA 46  
 43DB 1F 46  
 43DD 47  
 43DE 63 46  
 43E0 48  
 43E1 1F 46  
 43E3 49  
 43E4 72 46  
 43E6 4A  
 43E7 65 46  
 43E9 4B  
 43EA 98 46  
 43EC 4C  
 43ED 40 47  
 43EF 4D  
 43F0 A3 46  
 43F2 4F  
 43F3 8E 46  
 43F5 51  
 43F6 C0 47  
 43F8 52  
 43F9 3B 47  
 43FB 54  
 43FC DB 47  
 3FE 55  
 43FF D5 46  
 4401 56  
 4402 D9 46  
 4404 57  
 4405 FF 46  
 4407 58  
 4408 52 48  
 440A 5A  
 440B 69 48  
 440D 00  
 440E 00 00  
 4410 00  
 4411 00 00  
 4413 00  
 4414 00 00  
 4416 00  
 4417  
 4417 3A 00 41  
 441A CD B6 43  
 441D 21 00 41  
 4420 06 04  
 4422 EB  
 4423 13  
 4424 1A  
 4425 FE 0D  
 4427 CA 49 44  
 442A FE 20  
 442C CA 3E 44

2810	DW	CHEK
2820	DB	'D'
2830	DW	DUMP
2840	DB	'E'
2850	DW	EDIT
2860	DB	'F'
2870	DW	LOOK
2880	DB	'G'
2890	DW	EXCT
2900	DB	'H'
2910	DW	LOOK
2920	DB	'I'
2930	DW	ICMD
2940	DB	'J'
2950	DW	JUMP
2960	DB	'K'
2970	DW	ECHO
2980	DB	'L'
2990	DW	LDGO
3000	DB	'M'
3010	DW	MOVE
3020	DB	'O'
3030	DW	OUTCM
3040	DB	'Q'
3050	DW	CKSUM
3060	DB	'R'
3070	DW	INP
3080	DB	'T'
3090	DW	TEST
3100	DB	'U'
3110	DW	USER
3120	DB	'V'
3130	DW	VERFY
3140	DB	'W'
3150	DW	WRIT
3160	DB	'X'
3170	DW	XCHG
3180	DB	'Z'
3190	DW	ZERO
3200	DB	0
3210	DW	0
3220	DB	0
3230	DW	0
3240	DB	0
3250	DW	0
3260	DB	0
3270 *		
3280 DECOD	LDA	IBUF
3290	CALL	COMM
3300	LXI	H,IBUF
3310	MVI	B,4
3320	XCHG	
3330 NXT	INX	D
3340	LDAX	D
3350	CPI	13
3360	JZ	SETUP
3370	CPI	' '
3380	JZ	COUNT

RESERVE FOR 3  
COMMANDS

DECODE BUFFER

```

442F 1B
4430 3E 04
4432 B8
4433 CA 3E 44
4436 13
4437 1A
4438 CD 9B 42
443B C3 23 44
443E
443E 05
443F CA 49 44
4442 E5
4443 21 00 00
4446 C3 23 44
4449
4449 78
444A 01 00 00
444D 11 00 00
4450 FE 04
4452 CA 65 44
4455 FE 03
4457 CA 69 44
445A FE 02
445C CA 62 44
445F E5
4460 C1
4461 E1
4462 D1
4463 23
4464 C9
4465
4465 E5
4466 21 00 00
4469 E5
446A 06 10
446C 09
446D 06 00
446F D1
4470 C9
4471
4471 0A
4472 0D
4473 43 4F 4D 4D 41
      4E 44 3F
447B 0D
447C
447C FE 0D
447E C2 8E 44
4481 C5
4482 01 00 18
4485 0D
4486 C2 85 44
4489 05
448A C2 85 44
448D C1
448E D7
448F C9
4490

```

```

3390      DCX      D
3400      MVI      A,-4
3410      CMP      B
3420      JZ        COUNT
3430      INX      D
3440      LDAX     D
3450      CALL     AHX
3460      JMP      NXT
3470 *
3480 COUNT DCR      B
3490      JZ        SETUP
3500      PUSH     H
3510      LXI      H,0
3520      JMP      NXT
3530 *
3540 SETUP MOV      A,B
3550      LXI      B,0
3560      LXI      D,0
3570      CPI      4
3580      JZ        AD0
3590      CPI      3
3600      JZ        AD1
3610      CPI      2
3620      JZ        AD2
3630      PUSH     H
3640      POP      B
3650      POP      H
3660 AD2  POP      D
3670      INX      H
3680      RET
3690 *
3700 AD0  PUSH     H
3710      LXI      H,0
3720 AD1  PUSH     H
3730      MVI      B,10H
3740      DAD      B
3750      MVI      B,0
3760      POP      D
3770      RET
3780 *
3790 PR1  DB        10
3800      DB        13
3810      ASC      'COMMAND?'
3820      DB        13.
3830 *
3840 CRT  CPI      13
3850      JNZ     CRT1
3860      PUSH     B
3870      LXI      B,1800H
3880 WAIT DCR      C
3890      JNZ     WAIT
3900      DCR      B
3910      JNZ     WAIT
3920      POP      B
3930 CRT1 RST      2
3940      RET
3950 *

```

SETUP REGISTERS

CRT OUTPUT ROUTINE

DELAY TO SLOW  
DOWN DUMPS

```

4490 2B
4491 F5
4492 7D
4493 B4
4494 4F
4495 F1
4496 FE 03
4498 DA 9C 44
449B 4F
449C 23
449D CD C0 42
44A0 CD D8 42
44A3 06 30
44A5 0C
44A6 0D
44A7 C2 AF 44
44AA 11 FF FF
44AD 06 10
44AF 7A
44B0 B3
44B1 C8
44B2 7E
44B3 E6 7F
44B5 1B
44B6 0C
44B7 0D
44B8 CA DC 44
44BB FE 1D
44BD CA CA 44
44C0 FE 0D
44C2 CA D5 44
44C5 FE 20
44C7 D2 CC 44
44CA 3E 2E
44CC CD 7C 44
44CF 23
44D0 05
44D1 C2 AF 44
44D4 2B
44D5 23
44D6 CD 74 42
44D9 C3 A0 44
44DC
44DC FE 20
44DE DA F9 44
44E1 FE 1D
44E3 CA F9 44
44E6 CD 7C 44
44E9 7E
44EA 07
44EB D2 FF 44
44EE 3E 2E
44F0 CD 7C 44
44F3 CD E6 42
44F6 C3 CF 44
44F9
44F9 CD F3 42
44FC C3 F3 44

```

```

3960 ASCI DCX H
3970 PUSH PSW
3980 MOV A,L
3990 ORA H
4000 MOV C,A
4010 POP PSW
4020 CPI 3
4030 JC $+1
4040 MOV C,A
4050 INX H
4060 CALL DLMT
4070 ASC1 CALL ADRO
4080 MVI B,48
4090 INR C
4100 DCR C
4110 JNZ ASC2
4120 LXI D,-1
4130 MVI B,16
4140 ASC2 MOV A,D
4150 ORA E
4160 RZ
4170 MOV A,M
4180 ANI 7FH
4190 DCX D
4200 INR C
4210 DCR C
4220 JZ ASC5
4230 CPI 1DH
4240 JZ ASC3
4250 CPI 13
4260 JZ ASC4
4270 CPI 20H
4280 JNC $+2
4290 ASC3 MVI A,'.'
4300 CALL CRT
4310 ASC7 INX H
4320 DCR B
4330 JNZ ASC2
4340 DCX H
4350 ASC4 INX H
4360 CALL PAUSE
4370 JMP ASC1
4380 *
4390 ASC5 CPI 20H
4400 JC ASC6
4410 CPI 1DH
4420 JZ ASC6
4430 CALL CRT
4440 MOV A,M
4450 RLC
4460 JNC ASC9
4470 MVI A,'.'
4480 CALL CRT
4490 ASC8 CALL SP1
4500 JMP ASC7
4510 *
4520 ASC6 CALL OUTM
4530 JMP ASC8

```

ASCII DUMP (A)

```

44FF
44FF CD E3 42
4502 C3 CF 44
4505
  505 3E 2E
4507 C3 7C 44
450A
450A FE 03
450C DA 39 45
450F EB
4510 11 F0 D8
4513 CD 2B 45
4516 11 18 FC
4519 CD 2B 45
451C 11 9C FF
451F CD 2B 45
4522 11 F6 FF
4525 CD 2B 45
4528 11 FF FF
452B 3E 2F
452D E5
452E C1
452F 3C
4530 E5
4531 19
4532 DA 2E 45
4535 E1
4536 C3 7C 44
4539
  539 2B
+53A E5
453B 21 8E 45
453E CD AB 42
4541 CD B6 42
4544 E1
4545 E5
4546 CD C8 42
4549 CD E6 42
454C CD 88 45
454F EB
4550 E3
4551 CD 88 45
4554 EB
4555 E5
4556 CD C8 42
4559 C1
455A E5
455B 09
455C CD 88 45
455F EB
4560 CD 88 45
4563 22 34 41
4566 E1
4567 E3
4568 CD EB 42
456B E3
456C E5
456D 2A 34 41

```

```

4540 *
4550 ASC9 CALL SP2
4560 JMP ASC7
4570 *
4580 DOT MVI A, '.'
4590 JMP CRT
4600 *
4610 HEXAR CPI 3
4620 JC HEX2
4630 XCHG
4640 DECML LXI D, -10000
4650 CALL DIVD
4660 LXI D, -1000
4670 CALL DIVD
4680 LXI D, -100
4690 CALL DIVD
4700 DEC2 LXI D, -10
4710 CALL DIVD
4720 LXI D, -1
4730 DIVD MVI A, '0'-1
4740 PUSH H
4750 POP B
4760 INR A
4770 PUSH H
4780 DAD D
4790 JC DIVD+3
4800 POP H
4810 JMP CRT
4820 *
4830 HEX2 DCX H
4840 PUSH H
4850 LXI H, YMSG
4860 CALL STRNG
4870 CALL CRLF
4880 POP H
4890 PUSH H
4900 CALL LENG
4910 CALL SP1
4920 CALL HLSP2
4930 XCHG
4940 XTHL
4950 CALL HLSP2
4960 XCHG
4970 PUSH H
4980 CALL LENG
4990 POP B
5000 PUSH H
5010 DAD B
5020 CALL HLSP2
5030 XCHG
5040 CALL HLSP2
5050 SHLD YTEMP
5060 POP H
5070 XTHL
5080 CALL HLOUT
5090 XTHL
5100 PUSH H
5110 LHLD YTEMP

```

HEX ARITHMETIC (B)

DECIMAL OUTPUT



```

4570 E3
4571 D5
4572 E5
4573 C5
4574 CD B6 42
4577 06 06
4579 C5
457A C1
457B 05
457C C8
457D E1
457E C5
457F CD 10 45
4582 CD E6 42
4585 C3 7A 45
4588
4588 CD EB 42
458B C3 E3 42
458E
458E 1C
458F 20 20 41 20 20
      20 20 20 42 20
      20 20 20 41 2B
      42 20 20 20 41
      2D 42 20 20 20
      42 2D 41

```

```

45AB
45AB CD C0 42
45AE CD D8 42
45B1 06 10
45B3 CD F3 42
45B6 23
45B7 1B
45B8 7A
45B9 B3
45BA C8
45BB CD E6 42
45BE 05
45BF C2 B3 45
45C2 CD 74 42
45C5 C3 AE 45
45C8
45C8 1B
45C9 EB
45CA CD D8 42
45CD EB
45CE EB
45CF CD F3 42
45D2 EB
45D3 3E 2D
45D5 CD 7C 44
45D8 21 00 00
45DB 06 01
45DD CD 42 42
45E0 CA DD 45
45E3 4F
45E4 CD 8D 42
45E7 79

```

```

5120      XTHL
5130      PUSH D
5140      PUSH H
5150      PUSH B
5160      CALL CRLF
5170      MVI B,6
5180      PUSH B
5190 HEX1  POP B
5200      DCR B
5210      RZ
5220      POP H
5230      PUSH B
5240      CALL DECML
5250      CALL SP1
5260      JMP HEX1
5270 *
5280 HLSP2 CALL HLOUT
5290      JMP SP2
5300 *
5310 YMSG  DB 28
5320      ASC ' A B A+B A-B B-A'

```

```

5330 *
5340 DUMP  CALL DLMT
5350      CALL ADRO
5360      MVI B,16
5370 DM1   CALL OUTM
5380      INX H
5390      DCX D
5400      MOV A,D
5410      ORA E
5420      RZ
5430      CALL SP1
5440      DCR B
5450      JNZ DM1
5460      CALL PAUSE
5470      JMP DUMP+3
5480 *
5490      DCX D
5500 EDIT  XCHG
5510      CALL ADRO
5520      XCHG
5530 EDIT2 XCHG
5540      CALL OUTM
5550      XCHG
5560      MVI A,'-'
5570      CALL CRT
5580      LXI H,0
5590      MVI B,1
5600 EDIT3 CALL KEYBG
5610      JZ EDIT3
5620      MOV C,A
5630      CALL HEXC
5640      MOV A,C

```

DUMP HEX (D)

EDIT (E)

```

45E8 D2 09 46
45EB FE 20
45ED CA 13 46
45F0 FE 0D
45F2 CA 19 46
45F5 FE 1B
45F7 CA C8 45
45FA FE 03
45FC C2 DD 45
45FF 05
4600 CA 05 46
4603 7D
4604 12
4605 13
4606 C3 E3 42
4609
4609 CD 7C 44
460C 47
460D CD 9B 42
4610 C3 DD 45
4613
4613 CD FF 45
4616 C3 CE 45
4619
4619 CD FF 45
461C C3 C9 45
461F
461F CD C8 42
4622 7E
4623 B9
4624 CC 2F 46
4627 7A
4628 B3
4629 C8
462A 23
462B 1B
462C C3 22 46
462F
462F 3A 00 41
4632 FE 46
4634 C2 48 46
4637 CD D8 42
463A CD F3 42
463D CD E3 42
4640 23
4641 CD F3 42
4644 2B
4645 C3 74 42
4648
4648 23
4649 7E
464A B8
464B 2B
464C C0
464D 2B
464E CD D8 42
4651 CD F3 42
4654 CD E3 42

```

```

5650 JNC EDIT4
5660 CPI ' '
5670 JZ EDIT5
5680 CPI 13
5690 JZ EDIT6
5700 CPI 1BH
5710 JZ EDIT-1
5720 CPI 3
5730 JNZ EDIT3
5740 EDIT8 DCR B
5750 JZ EDIT9
5760 MOV A,L
5770 STAX D
5780 EDIT9 INX D
5790 JMP SP2
5800 *
5810 EDIT4 CALL CRT
5820 MOV B,A
5830 CALL AHX
5840 JMP EDIT3
5850 *
5860 EDIT5 CALL EDIT8
5870 JMP EDIT2
5880 *
5890 EDIT6 CALL EDIT8
5900 JMP EDIT
5910 *
5920 LOOK CALL LENG
5930 MOV A,M
5940 CMP C
5950 CZ LK2
5960 MOV A,D
5970 ORA E
5980 RZ
5990 INX H
6000 DCX D
6010 JMP LOOK+3
6020 *
6030 LK2 LDA IBUF
6040 CPI 'F'
6050 JNZ LK3
6060 CALL ADRO
6070 CALL OUTM
6080 CALL SP2
6090 INX H
6100 CALL OUTM
6110 DCX H
6120 JMP PAUSE
6130 *
6140 LK3 INX H
6150 MOV A,M
6160 CMP B
6170 DCX H
6180 RNZ
6190 DCX H
6200 CALL ADRO
6210 CALL OUTM
6220 CALL SP2

```

1,2 BYTE SEARCH (F,H)

4657 23  
 4658 23  
 4659 CD F3 42  
 465C 2B  
 65D CD F3 42  
 4660 C3 74 42  
 4663  
 4663 D5  
 4664 C9  
 4665  
 4665 FE 04  
 4667 CA 6E 46  
 466A EB  
 466B 22 3A 41  
 466E 2A 3A 41  
 4671 E9  
 4672  
 4672 FE 04  
 4674 CA 0B 43  
 4677 53  
 4678 1E DB  
 467A 3E CD  
 467C 01 F4 42  
 467F 65  
 4680 68  
 4681 41  
 4682 4F  
 4683 7C  
 4684 26 C3  
 4686 E5  
 4687 C5  
 4688 D5  
 4689 21 00 00  
 468C 39  
 468D E9  
 468E  
 468E 53  
 468F 1E D3  
 4691 2D  
 4692 AF  
 4693 47  
 4694 4F  
 4695 C3 7F 46  
 4698  
 4698 3E 04  
 469A CD 7C 44  
 469D CD 83 42  
 46A0 C3 9A 46  
 46A3  
 46A3 FE 02  
 46A5 D2 D0 42  
 46A8 E5  
 46A9 CD C8 42  
 46AC D5  
 46AD EB  
 46AE C5  
 46AF E1  
 46B0 CD C8 42

6230 INX H  
 6240 INX H  
 6250 CALL OUTM  
 6260 DCX H  
 6270 CALL OUTM  
 6280 JMP PAUSE  
 6290 \*  
 6300 EXCT PUSH D  
 6310 RET  
 6320 \*  
 6330 JUMP CPI 4  
 6340 JZ \$+4  
 6350 XCHG  
 6360 SHLD GOLOC  
 6370 G1 LHLD GOLOC  
 6380 PCHL  
 6390 \*  
 6400 ICMD CPI 4  
 6410 JZ INITC  
 6420 MOV D,E  
 6430 MVI E,0DBH  
 6440 MVI A,0CDH  
 6450 LXI B,OUTH  
 6460 ICMD1 MOV H,L  
 6470 MOV L,B  
 6480 MOV B,C  
 6490 MOV C,A  
 6500 MOV A,H  
 6510 MVI H,0C3H  
 6520 PUSH H  
 6530 PUSH B  
 6540 PUSH D  
 6550 LXI H,0  
 6560 DAD SP  
 6570 PCHL  
 6580 \*  
 6590 OUTCM MOV D,E  
 6600 MVI E,0D3H  
 6610 DCR L  
 6620 XRA A  
 6630 MOV B,A  
 6640 MOV C,A  
 6650 JMP ICMD1  
 6660 \*  
 6670 ECHO MVI A,4  
 6680 CALL CRT  
 6690 CALL KEYB  
 6700 JMP ECHO+2  
 6710 \*  
 6720 MOVE CPI 2  
 6730 JNC ERR  
 6740 PUSH H  
 6750 CALL LENG  
 6760 PUSH D  
 6770 XCHG  
 6780 PUSH B  
 6790 POP H  
 6800 CALL LENG

EXECUTE (G)

GO (J)

"I" COMMAND (I)  
 INITIALIZE OR  
 INPUT FROM PORT

PORT OUTPUT (O)

KEYBOARD ECHO (K)

MOVE MEMORY (M)

46B3 D2 C3 46  
 46B6 D1  
 46B7 F1  
 46B8 7E  
 46B9 02  
 46BA 03  
 46BB 23  
 46BC 1B  
 46BD 7A  
 46BE B3  
 46BF C2 B8 46  
 46C2 C9  
 46C3  
 46C3 D1  
 46C4 D5  
 46C5 E1  
 46C6 09  
 46C7 C1  
 46C8 2B  
 46C9 0B  
 46CA 0A  
 46CB 77  
 46CC 0B  
 46CD 2B  
 46CE 1B  
 46CF 7A  
 46D0 B3  
 46D1 C2 CA 46  
 46D4 C9  
 6D5  
 46D5 2B  
 46D6 C3 80 48  
 46D9  
 46D9 FE 02  
 46DB D2 D0 42  
 46DE CD C8 42  
 46E1 0A  
 46E2 BE  
 46E3 C4 EF 46  
 46E6 23  
 46E7 03  
 46E8 1B  
 46E9 7A  
 46EA B3  
 46EB C2 E1 46  
 46EE C9  
 46EF  
 46EF CD D8 42  
 46F2 CD F3 42  
 46F5 CD E3 42  
 46F8 0A  
 46F9 CD F4 42  
 46FC C3 74 42  
 46FF  
 46FF F5  
 4700 CD C8 42  
 4703 F1  
 4704 E5

6810 JNC GTR  
 6820 POP D  
 6830 POP PSW  
 6840 MVL MOV A,M  
 6850 STAX B  
 6860 INX B  
 6870 INX H  
 6880 DCX D  
 6890 MOV A,D  
 6900 ORA E  
 6910 JNZ MVL  
 6920 RET  
 6930 \*  
 6940 GTR POP D  
 6950 PUSH D  
 6960 POP H  
 6970 DAD B  
 6980 POP B  
 6990 DCX H  
 7000 DCX B  
 7010 MV2 LDAX B  
 7020 MOV M,A  
 7030 DCX B  
 7040 DCX H  
 7050 DCX D  
 7060 MOV A,D  
 7070 ORA E  
 7080 JNZ MV2  
 7090 RET  
 7100 \*  
 7110 USER DCX H  
 7120 JMP RAM  
 7130 \*  
 7140 VRFY CPI 2  
 7150 JNC ERR  
 7160 CALL LENG  
 7170 VR1 LDAX B  
 7180 CMP M  
 7190 CNZ VR2  
 7200 INX H  
 7210 INX B  
 7220 DCX D  
 7230 MOV A,D  
 7240 ORA E  
 7250 JNZ VR1  
 7260 RET  
 7270 \*  
 7280 VR2 CALL ADRO  
 7290 CALL OUTM  
 7300 CALL SP2  
 7310 LDAX B  
 7320 CALL OUTH  
 7330 JMP PAUSE  
 7340 \*  
 7350 WRIT PUSH PSW  
 7360 CALL LENG  
 7370 POP PSW  
 7380 PUSH H

USER (U)

VERIFY MEMORY (V)

WRITE TAPE (W)

4705 FE 02  
 4707 D2 0C 47  
 470A C5  
 470B E1  
 70C CD 65 42  
 470F 7C  
 4710 CD 71 42  
 4713 7D  
 4714 CD 71 42  
 4717 E1  
 4718 7A  
 4719 CD 71 42  
 471C 7B  
 471D CD 71 42  
 4720 0E 00  
 4722 7E  
 4723 CD 71 42  
 4726 7E  
 4727 CD 85 47  
 472A C2 22 47  
 472D 79  
 472E 41  
 472F CD 71 42  
 4732 48  
 4733 C3 CF 47  
 4736  
 4736 0E 00  
 4738 C3 42 47  
 473B  
 73B 0E 01  
 473D C3 42 47  
 4740  
 4740 0E 02  
 4742 2B  
 4743 F5  
 4744 AF  
 4745 F5  
 4746 CD 74 42  
 4749 F1  
 474A CD 81 0F  
 474D FE A5  
 474F C2 45 47  
 4752 F1  
 4753 FE 04  
 4755 CC AD 47  
 4758 FE 03  
 475A C2 62 47  
 475D D5  
 475E CD AD 47  
 4761 E1  
 4762 E5  
 4763 06 00  
 4765 CD 5C 42  
 4768 0C  
 769 0D  
 476A CA 6E 47  
 476D 77  
 476E CD 83 47

7390 CPI 2  
 7400 JNC \$+2  
 7410 PUSH B  
 7420 POP H  
 7430 CALL PREP  
 7440 MOV A,H  
 7450 CALL COUT  
 7460 MOV A,L  
 7470 CALL COUT  
 7480 POP H  
 7490 MOV A,D  
 7500 CALL COUT  
 7510 MOV A,E  
 7520 CALL COUT  
 7530 WR1 MVI C,0  
 7540 WR2 MOV A,M  
 7550 CALL COUT  
 7560 MOV A,M  
 7570 CALL ENDCK+2  
 7580 JNZ WR2  
 7590 MOV A,C  
 7600 MOV B,C  
 7610 CALL COUT  
 7620 IN6 MOV C,B  
 7630 JMP CKSM2  
 7640 \*  
 7650 CHEK MVI C,0  
 7660 JMP LDGO+2  
 7670 \*  
 7680 INP MVI C,1  
 7690 JMP LDGO+2  
 7700 \*  
 7710 LDGO MVI C,2  
 7720 DCX H  
 7730 PUSH PSW  
 7740 XRA A  
 7750 SYNC PUSH PSW  
 7760 CALL PAUSE  
 7770 POP PSW  
 7780 CALL CIN1  
 7790 CPI 0A5H  
 7800 JNZ SYNC  
 7810 POP PSW  
 7820 CPI 4  
 7830 CZ R4  
 7840 CPI 3  
 7850 JNZ IN2  
 7860 PUSH D  
 7870 CALL R4  
 7880 POP H  
 7890 IN2 PUSH H  
 7900 MVI B,0  
 7910 IN3 CALL CIN  
 7920 INR C  
 7930 DCR C  
 7940 JZ \$+1  
 7950 MOV M,A  
 7960 CALL ENDCK

CHECK TAPE (C)

READ TAPE (R)

LOAD AND GO (L)

4771 C2 65 47  
 4774 E1  
 4775 CD 5C 42  
 4778 B8  
 4779 C2 8A 47  
 477C 79  
 477D FE 02  
 477F C2 32 47  
 4782 E9  
 4783  
 4783 80  
 4784 47  
 4785 23  
 4786 1B  
 4787 7A  
 4788 B3  
 4789 C9  
 478A  
 478A F5  
 478B 48  
 478C 21 9C 47  
 478F CD AB 42  
 4792 F1  
 4793 CD F4 42  
 4796 CD E3 42  
 4799 C3 D7 47  
 479C  
 479C 10  
 479D 43 48 45 43 4B  
 53 55 4D 20 45  
 52 52 4F 52 20  
 20

47AD  
 47AD F5  
 47AE CD 5C 42  
 47B1 67  
 47B2 CD 5C 42  
 47B5 6F  
 47B6 CD 5C 42  
 47B9 57  
 47BA CD 5C 42  
 47BD 5F  
 47BE F1  
 47BF C9  
 47C0  
 47C0 CD C8 42  
 47C3 0E 00  
 47C5 79  
 47C6 86  
 47C7 4F  
 47C8 23  
 47C9 1B  
 47CA 7A  
 47CB B3  
 47CC C2 C4 47  
 47CF 06 09  
 47D1 21 9C 47  
 47D4 CD AC 42

7970 JNZ IN3  
 7980 POP H  
 7990 CALL CIN  
 8000 CMP B  
 8010 JNZ IN5  
 8020 MOV A,C  
 8030 CPI 2  
 8040 JNZ IN6  
 8050 PCHL  
 8060 \*  
 8070 ENDCK ADD B  
 8080 MOV B,A  
 8090 INX H  
 8100 DCX D  
 8110 MOV A,D  
 8120 ORA E  
 8130 RET  
 8140 \*  
 8150 IN5 PUSH PSW  
 8160 MOV C,B  
 8170 LXI H,CERR  
 8180 CALL STRNG  
 8190 POP PSW  
 8200 CALL OUTH  
 8210 CALL SP2  
 8220 JMP CKSM3  
 8230 \*  
 8240 CERR DB 16  
 8250 ASC 'CHECKSUM ERROR'

8260 \*  
 8270 R4 PUSH PSW  
 8280 CALL CIN  
 8290 MOV H,A  
 8300 CALL CIN  
 8310 MOV L,A  
 8320 CALL CIN  
 8330 MOV D,A  
 8340 CALL CIN  
 8350 MOV E,A  
 8360 POP PSW  
 8370 RET  
 8380 \*  
 8390 CKSUM CALL LENG  
 8400 MVI C,0  
 8410 MOV A,C  
 8420 ADD M  
 8430 MOV C,A  
 8440 INX H  
 8450 DCX D  
 8460 MOV A,D  
 8470 ORA E  
 8480 JNZ CKSUM+4  
 8490 CKSM2 MVI B,9  
 8500 LXI H,CERR  
 8510 CALL STRNG+1

CHECKSUM (Q)

47D7 79  
 47D8 C3 F4 42  
 47DB  
 47DB CD C8 42  
 47DE 01 5A 5A  
 47E1 CD 27 48  
 47E4 C5  
 47E5 E5  
 47E6 D5  
 47E7 CD 27 48  
 47EA 70  
 47EB 23  
 47EC 1B  
 47ED 7A  
 47EE B3  
 47EF C2 E7 47  
 47F2 CD 74 42  
 47F5 D1  
 47F6 E1  
 47F7 C1  
 47F8 E5  
 47F9 D5  
 47FA CD 27 48  
 47FD 7E  
 47FE B8  
 47FF C4 15 48  
 4802 23  
 4803 1B  
 4804 7A  
 4805 B3  
 4806 C2 FA 47  
 4809 21 32 41  
 480C 34  
 480D CC 36 48  
 4810 D1  
 4811 E1  
 4812 C3 E1 47  
 4815  
 4815 F5  
 4816 CD 74 42  
 4819 CD D8 42  
 481C 78  
 481D CD F4 42  
 4820 CD E3 42  
 4823 F1  
 4824 C3 F4 42  
 4827  
 4827 78  
 4828 E6 B4  
 482A A7  
 482B EA 2F 48  
 482E 37  
 482F 79  
 4830 17  
 4831 4F  
 4832 78  
 4833 17  
 4834 47

8520 CKSM3 MOV A,C  
 8530 JMP OUTH  
 8540 \*  
 8550 TEST CALL LENG  
 8560 LXI B,5A5AH  
 8570 CYCL CALL RNDM  
 8580 PUSH B  
 8590 PUSH H  
 8600 PUSH D  
 8610 TLOP CALL RNDM  
 8620 MOV M,B  
 8630 INX H  
 8640 DCX D  
 8650 MOV A,D  
 8660 ORA E  
 8670 JNZ TLOP  
 8680 CALL PAUSE  
 8690 POP D  
 8700 POP H  
 8710 POP B  
 8720 PUSH H  
 8730 PUSH D  
 8740 RLOP CALL RNDM  
 8750 MOV A,M  
 8760 CMP B  
 8770 CNZ MERR  
 8780 INX H  
 8790 DCX D  
 8800 MOV A,D  
 8810 ORA E  
 8820 JNZ RLOP  
 8830 LXI H,TCNTR  
 8840 INR M  
 8850 CZ TESTN  
 8860 POP D  
 8870 POP H  
 8880 JMP CYCL  
 8890 \*  
 8900 MERR PUSH PSW  
 8910 CALL PAUSE  
 8920 CALL ADRO  
 8930 MOV A,B  
 8940 CALL OUTH  
 8950 CALL SP2  
 8960 POP PSW  
 8970 JMP OUTH  
 8980 \*  
 8990 RNDM MOV A,B  
 9000 ANI 0B4H  
 9010 ANA A  
 9020 JPE PEVE  
 9030 STC  
 9040 PEVE MOV A,C  
 9050 RAL  
 9060 MOV C,A  
 9070 MOV A,B  
 9080 RAL  
 9090 MOV B,A

MEMORY TEST (T)

ERROR FOUND

PSEUDO-RANDOM  
NUMBER GENERATOR

4835 C9  
 4836  
 4836 3A 33 41  
 4839 C6 01  
 483B 27  
 483C 32 33 41  
 483F CD F4 42  
 4842 21 4B 48  
 4845 CD AB 42  
 4848 C3 B6 42  
 484B  
 484B 06  
 484C 20 54 45 53 54  
 53  
 4852  
 4852 FE 02  
 4854 D2 D0 42  
 4857 CD C8 42  
 485A 0A  
 485B F5  
 485C 7E  
 485D 02  
 485E F1  
 485F 77  
 4860 23  
 4861 03  
 4862 1B  
 4863 7A  
 4864 B3  
 4865 C2 5A 48  
 4868 C9  
 4869  
 4869 CD C8 42  
 486C 71  
 486D 23  
 486E 1B  
 486F 7A  
 4870 B3  
 4871 C2 6C 48  
 4874 C9  
 4875  
 4875  
 4875  
 4880 C9  
 4881

9100 RET  
 9110 \*  
 9120 TESTN LDA DECNB  
 9130 ADI 1  
 9140 DAA  
 9150 STA DECNB  
 9160 CALL OUTH  
 9170 LXI H,NMSG  
 9180 CALL STRNG  
 9190 JMP CRLF  
 9200 \*  
 9210 NMSG DB 6  
 9220 ASC ' TESTS'  
 9230 \*  
 9240 XCHG CPI 2  
 9250 JNC ERR  
 9260 CALL LENG  
 9270 XCH1 LDAX B  
 9280 PUSH PSW  
 9290 MOV A,M  
 9300 STAX B  
 9310 POP PSW  
 9320 MOV M,A  
 9330 INX H  
 9340 INX B  
 9350 DCX D  
 9360 MOV A,D  
 9370 ORA E  
 9380 JNZ XCH1  
 9390 RET  
 9400 \*  
 9410 ZERO CALL LENG  
 9420 MOV M,C  
 9430 INX H  
 9440 DCX D  
 9450 MOV A,D  
 9460 ORA E  
 9470 JNZ ZERO+3  
 9480 RET  
 9490 \*  
 9500 ZZZ EQU \$  
 9510 \*  
 9520 ORG RAM  
 9530 DB 0C9H  
 9540 \*

TEST COUNTER

EXCHANGE MEMORY (X)

FILL MEMORY (Z)

END OF FILE

USER LOCATION

SYMBOL TABLE



# SYMBOL TABLE <sup>ASC2</sup> <sub>ASC8</sub>

AD0	4465	AD1	4469	AD2	4462	ADRO	42D8	AHEX	429B	ASC1	44A0
ASC2	44AF	ASC3	44CA	ASC4	44D5	ASC5	44DC	ASC6	44F9	ASC7	44CF
ASC8	44F3	ASC9	44FF	ASCI	4490	BIASC	42FD	BLANK	437B	BS1	4373
BS2	4381	BUFL	003E	CERR	479C	CHEK	4736	CIN	425C	CIN1	0F81
CIN8	0FA1	CKSM2	47CF	CKSM3	47D7	CKSUM	47C0	CMND	4345	CMSG	4215
COM1	43BD	COMM	43B6	COUNT	443E	COUT	4271	COUT8	0FA9	CRLF	42B6
CRT	447C	CRT1	448E	CTBLE	43CB	CYCL	47E1	DEC2	4522	DECML	4510
DECNB	4133	DECOD	4417	DIVD	452B	DLMT	42C0	DM1	45B3	DOT	4505
DUMP	45AB	ECHO	4698	EDIT	45C9	EDIT2	45CE	EDIT3	45DD	EDIT4	4609
EDIT5	4613	EDIT6	4619	EDIT8	45FF	EDIT9	4605	ENDCK	4783	ERASE	43B1
ERR	42D0	EXCT	4663	FLGP	4131	FULL	43A8	G1	466E	GOLOC	413A
GTR	46C3	HEX1	457A	HEX2	4539	HEXAR	450A	HEXC	428D	HLOUT	42EB
HLSP2	4588	IBUF	4100	ICMD	4672	ICMD1	467F	IFLAG	4130	IN2	4762
IN3	4765	IN5	478A	IN6	4732	INIT	4316	INIT2	4319	INITC	430B
INP	473B	JUMP	4665	KEYB	4283	KEYBC	4252	KEYBG	4242	LDGO	4740
LENG	42C8	LK2	462F	LK3	4648	LOOK	461F	MERR	4815	MONI	4203
MOVE	46A3	MV1	46B8	MV2	46CA	NMSG	484B	NUMB	4306	NXT	4423
OUTCM	468E	OUTH	42F4	OUTM	42F3	PAUSE	4274	PEVE	482F	PR1	4471
PREP	4265	PRMPT	4338	R4	47AD	RAM	4880	RDPTR	4136	READ	4386
RLESE	0B4C	RLOP	47FA	RNDM	4827	SCAN	0B55	SCPAD	4100	SETUP	4449
SP1	42E6	SP2	42E3	STAK	4200	START	430E	STAT	387F	STRNG	42AB
SYNC	4745	TCNTR	4132	TEST	47DB	TESTN	4836	TLOP	47E7	USER	46D5
VR1	46E1	VR2	46EF	VRFY	46D9	WAIT	4485	WR1	4720	WR2	4722
WRIT	46FF	WRPTR	4138	XCH1	485A	XCHG	4852	YMSG	458E	YTEMP	4134
ZERO	4869	ZZZ	4875								